计算机试验班 61 秦佳璐 2140506071 汇编语言程序设计作业

五六七八九章

第五章循环分支程序设计

5-3 将 AX 寄存器中的 16 位数分成 4 组，每组 4 位，然后把这四组数分别放在 AL、BL、CL 和 DL 中。

```
DATAS SEGMENT
    store db 4 dup(?)
    ;此处输入数据段代码
DATAS ENDS
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
START:
    MOV AX,DATAS
    MOV DS,AX

    MOV    CL, 4;右移四次
   MOV CH, 4;循环四次
   LEA BX, STORE
A10:
   MOV DX, AX
   AND DX, 0FH    ;取AX的低四位
   MOV [BX], DL   ;低四位存入STORE中
   INC BX
   SHR AX, CL     ;右移四次
   DEC CH
   JNZ A10        ;循环四次完了码?
B10:
   MOV DL, STORE  ;四组数分别放在AL、BL、CL和DL中
   MOV CL, STORE+1
   MOV BL, STORE+2
   MOV AL, STORE+3

    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

5-7 试编写一个汇编语言程序，求出首地址为 DATA 的 100D 字数组中的最小偶数，并把它存放在 AX 中。

```
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    MOV    BX, 0
   MOV CX, 100
COMP:
   MOV AX, DATA[BX];取数组的第一个偶数
   ADD BX, 2
   TEST AX, 01H  ;是偶数吗?
   LOOPNZ COMP   ;不是，比较下一个数
   JNZ STOP      ;没有偶数，退出
   JCXZ STOP     ;最后一个数是偶数，即为最小偶数，退出
COMP1:
   MOV DX, DATA[BX];取数组的下一个偶数
   ADD BX, 2
   TEST DX, 01H  ;是偶数吗?
   JNZ NEXT      ;不是，比较下一个数
   CMP AX, DX        ;(AX)<(DX)吗?
   JLE NEXT
   MOV AX, DX    ;(AX)<(DX)，则置换(AX)为最小偶数
NEXT:
   LOOP   COMP1
STOP:
   RET

    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

5-13 在 STRING 到 STRING+99 单元中存放着一个字符串，试编制一个程序测试该字符串中是否存在数字，如有则把 CL 的第 5 位置 1，否则将该位置 0。

```
DATAS SEGMENT
    STRING DB DUP(?)
    ;此处输入数据段代码
DATAS ENDS
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
START:
    MOV AX,DATAS
    MOV DS,AX
```

```
    PUSH DS
    SUB AX, AX
    PUSH AX
    MOV AX, DSEG
    MOV DS, AX

BEGIN:
    MOV SI, 0
    MOV CX, 100
REPEAT:
    MOV AL, STRING [SI]
    CMP AL, 30H
    JB  GO_ON
    CMP AL, 39H
    JA  GO_ON
    OR  CL, 20H
    JMP EXIT
GO_ON:
    INC SI
    LOOP REPEAT
    AND CL, 0DFH
EXIT:
    RET

    ;此处输入代码段代码
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

5-14 在首地址为 TABLE 的数组中按递增次序存放着 100H 个 16 位补码数，试编写一个程序把出现次数最多的数及其出现次数分别存放于 AX 和 CX 中。

```
DATAS SEGMENT
    ;此处输入数据段代码
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
```

```
MAIN PROC FAR
    MOV AX,DATAS
    MOV DS,AX

START:
    PUSH DS
    SUB AX, AX
    PUSH AX
    MOV AX, DSEG
    MOV DS, AX
BEGIN:
    MOV CX, 100H
    MOV SI, 0
NEXT:
    MOV DX, 0
    MOV AX, TABLE [SI]
COMP:
    CMP TABLE [SI], AX    ;计算一个数的出现次数
    JNE ADDR
    INC DX
    ADD SI, 2
    LOOP COMP
ADDR:
    CMP DX, COUNT
    JLE DONE
    MOV COUNT, DX
    MOV DATA, AX
DONE:
    LOOP NEXT
    MOV CX, COUNT
    MOV AX, DATA
    RET

    ;此处输入代码段代码
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

5-15 数据段中已定义了一个有 n 个字数据的数组 M，试编写一程序求出 M 中绝对值最大的数，把它放在数据段的 M+2n 单元中，并将该数的偏移地址存放在 M+2(n+1)单元中。

```
DATA1 SEGMENT
    M dw 5,2,6,-3,10
    data dw ?;存数据
```

```
    adr dw ?;存地址
DATA1 ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATA1
START:
    MOV AX,DATA1
    MOV DS,AX

    lea si,M;取地址
    mov ax,[si];取值
    mov adr,si

    mov cx,5
    dec cx
    add si, 2

    cmp ax,0
    jns abs
    neg ax

abs:
    mov bx,[si]
    cmp bx,0
    JNS comp
    neg bx

comp:
    cmp ax,bx
    JAE getnum
    mov ax,bx
    mov adr,si

getnum:
    add si,2
    loop abs
    mov data,ax


    MOV AX,4C00H
    INT 21H
CODES ENDS
    END START
```

5-19 已知数组 A 包含 15 个互不相等的整数，数组 B 包含 20 个互不相等的整数。试编制一程序把既在 A 中又在 B 中出现的整数存放于数组 C 中。

```
DATAS SEGMENT
    A dw 0,2,4,6,8,10,12,14,16,18,20,22,24,26,28
    B dw 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
    cc dw 15 dup(?)
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    mov si,0
    mov bx,0
    mov cx,15

loop1:
   push cx
   mov cx,20
   mov di,0
   mov ax,A[si]

loop2:
   cmp B[di],ax
   JNZ continue
   mov cc[bx],ax;相等的时候move
   add bx,2

continue:
   add di,2
   loop loop2
   add si,2
   pop cx
   loop loop1

    MOV AH,4CH
    INT 21H
CODES ENDS
```

```
    END START
```

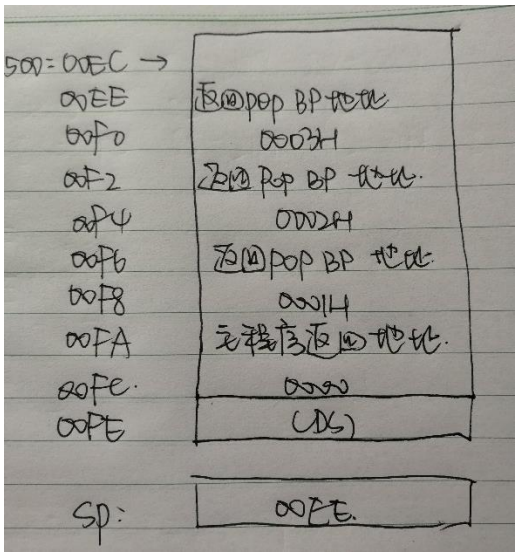
5-24（假设键盘输入的歌曲编号已经在 AL 中）假设已编制好 5 个歌曲程序，它们的段地址和偏移地址存放在数据段的跳跃表 SINGLIST 中。试编制一程序，根据从键盘输入的歌曲编号 1~5，转去执行五个歌曲程序中的某一个。

```
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MAIN PROC FAR
START:
    PUSH DS
    SUB AX, AX
    PUSH AX
    MOV AX, DSEG
    MOV DS, AX

BEGIN:
    MOV AH, 1
    INT 21H
    CMP AL, 0DH
    JZ EXIT
    SUB AL, '1'
    JB ERROR
    CMP AL, 4
    JA ERROR
    MOV BX, OFFSET SINGLIST
    MUL AX, 4
    ADD BX, AX
    JMP DWORD PTR[BX]
ERROR:
    MOV DX, OFFSET ERRMSG
    MOV AH, 09H
    INT 21H
    JMP BEGIN
SING1:    ┆
    JMP BEGIN
SING2:    ┆
    JMP BEGIN
SING3:    ┆
    JMP BEGIN
SING4:    ┆
    JMP BEGIN
SING5:    ┆
    JMP BEGIN
```

```
EXIT:
    RET
MAIN ENDP

    ;此处输入代码段代码
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

第六章 子程序结构

6-4 分析下面的程序，写出堆栈最满时各单元的地址及内容。



6-6 下面使用 STRUC 伪操作定义的参数表 NAMELIST(1)请用结构预置语句分配次结构的存储区（2）编写一段程序，从键盘输入字符（用 DOS 功能调用）存入结构中，然后将输入的字符送入 DISPFILE 单元中

（1） variable namelist <>

（2） mov ah,01
int 21h
mov displist, al

6-9 设有 10 个学生的成绩分别是 76，69，84，90，73，88，99，63，100 和 80 分。试编制一个子程序统计 60~69 分，70~79 分，80~89 分，90~99 分和 100 分的人数，分别存放到 S6，S7，S8，S9 和 S10 单元中。

```
DATAS SEGMENT
    SINGLIST  DD   SING1
DD  SING2
DD  SING3
DD  SING4
DD  SING5
```

```
ERRMSG DB  'Error! Invalid parameter!', 0DH, 0AH, '$'

    DATAS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
START:
MAIN PROC FAR
START:
    PUSH DS
    SUB AX, AX
    PUSH AX
    MOV AX, DSEG
    MOV DS, AX

BEGIN:
    MOV AH, 1
    INT 21H
    CMP AL, 0DH
    JZ EXIT
    SUB AL, '1'
    JB ERROR
    CMP AL, 4
    JA ERROR
    MOV BX, OFFSET SINGLIST
    MUL AX, 4
    ADD BX, AX
    JMP DWORD PTR[BX]
ERROR:
    MOV DX, OFFSET ERRMSG
    MOV AH, 09H
    INT 21H
    JMP BEGIN
SING1:     ┆
    JMP BEGIN
SING2:     ┆
    JMP BEGIN
SING3:     ┆
    JMP BEGIN
SING4:     ┆
    JMP BEGIN
SING5:     ┆
    JMP BEGIN
EXIT:
```

```
    RET
MAIN ENDP

    ;此处输入代码段代码
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

6-13 给定一个整数 N≥1,存放在 num 单元中，试编写一段递归子程序计算 fib(N)，并将结果存入 RESULT 单元中。Fibonacci 数的定义如下:

Fib(1) = 1

Fib(2) = 2

Fib(n) = Fib(n - 2) + Fib(n - 1) n > 2

```
DATAS SEGMENT
    ;此处输入数据段代码
    num dw 14
    result dw ?
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    ;此处输入代码段代码

    mov ax,num
    call fib
    mov result,bx
    MOV AH,4CH
    INT 21H

fib proc near
    push ax
    push cx

    cmp ax,2
    jle less_equal_2
    cmp ax,1
```

```
    jle less_equal_1

    dec ax
    call fib
    mov cx,bx
    dec ax
    call fib
    add bx,cx

    jmp ender

less_equal_2:
    mov bx,1
less_equal_1:
    mov bx,1

ender:
    pop cx
    pop ax
    mov result,bx
    ret

fib endp

CODES ENDS
    END START
```

第七章 高级汇编语言技术

7-3(1) 给定宏定义如下：(注意：此宏指令的功能是 V3←|V1-V2|)

```
    DIF         MACRO  X, Y
                MOV    AX, X
                SUB    AX, Y
                ENDM
    ABSDIF      MACRO  V1, V2, V3
                LOCAL  CONT
                PUSH   AX
                DIF    V1, V2
                CMP    AX, 0
                JGE    CONT
                NEG    AX
     CONT:      MOV    V3, AX
                POP    AX
                ENDM
```

试展开以下调用，并判定调用是否有效。

(1) ABSDIF P1, P2, DISTANCE

此宏定义有效，展开如下：

```
        PUSH AX
        DIFP1, P2
        MOV AX, P1
        SUB  AX, P2
        CMP  AX, 0
        JGE   M
        NEG  AX
M:
        MOV DISTANCE, AX
        POP  AX
```

7-4（源字符串首地址=arrys，目的字符串首地址=arryd）试编制宏定义，要求把存储器中的一个用 EOT（ASCII 码 04H）字符结尾的字符串传送到另一个存储区去。

```
SEND  MACRO SCHARS, DCHARS
    SEND  MACRO SCHARS, DCHARS
    LOCAL  NEXT, EXIT
    PUSH  AX
    PUSH  SI
    MOV SI, 0
NEXT:
    MOV AL, SCHARS[SI]
    MOV DCHARS[SI], AL
    CMP AL, 04H
    JZ  EXIT
    INC SI
    JMP NEXT
EXIT:
    POP SI
    POP AX
ENDM
```

7-7 下面的宏指令 CNT 和 INC1 完成相继字存储。

```
    CNT         MACRO  A, B
                A&B      DW   ?
                ENDM
    INC1        MACRO  A, B
                CNT      A, %B
      B=B+1
                ENDM
```

请展开下列宏调用：

C=0

```
        INC1    DATA, C
        INC1    DATA, C
```

展开:

```
C=0
INC1 DATA, C
DATA0   DW   ?
INC1 DATA, C
DATA0   DW   ?
```

7-8 定义宏指令并展开宏调用。宏指令 JOE 把一串信息‘MESSAGE NO. K’存入数据存储区 XK 中。宏调用为:

```
I=0
JOE     TEXT, I
        ⋮
JOE     TEXT, I
        ⋮
JOE     TEXT, I
```

定义:

```
MARY        MACRO  X, K
            X&KDB   ‘MESSAGE NO. &K’
            ENDM
JOE         MACRO  A, I
            MARY    A, %I
I=I+1
            ENDM
```

展开调用:

```
I=0
JOE     TEXT, I
TEXT0   DB   ‘MESSAGE NO. 0’
            ⋮
JOE     TEXT, I
TEXT1   DB   ‘MESSAGE NO. 1’
            ⋮
JOE     TEXT, I
TEXT2   DB   ‘MESSAGE NO. 2’
```

7-11 试编写一段程序完成以下功能，如给定名为 X 的字符串长度大于 5 时，下列指令将汇编 10 次。

```
    ADD     AX, AX
DATAS SEGMENT
    ;此处输入数据段代码
    xx db ’ ssasdh’
    nn db ’ ’
DATAS ENDS
```

```
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
START:
    INT 21H
    if nn-xx GT 5
       rept 10
          add ax,ax
       endm
    endif

    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

第八章 输入输出程序设计

8-1 写出分配给下列中断类型号在中断向量表中的物理地址。

(1) INT 12H 48-4B

(2) INT 8 20-23

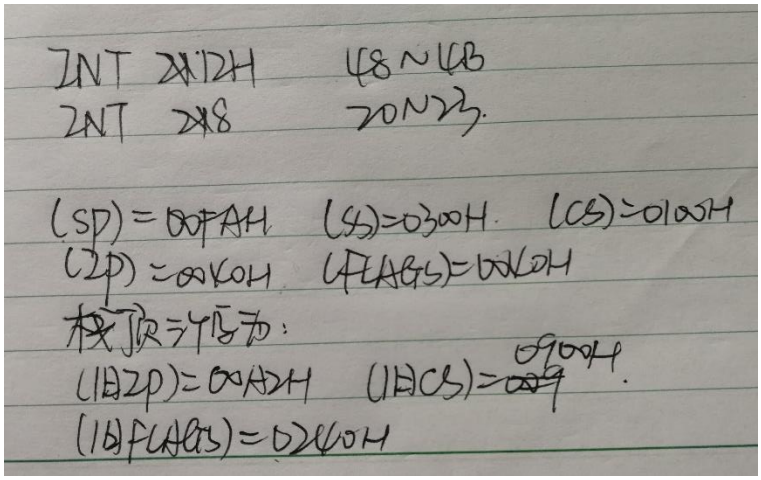
8-6 试编写程序，它轮流测试两个设备的状态寄存器，只要一个状态寄存器的第 0 位为 1，则就与其相应的设备输入一个字符；如果其中任一状态寄存器的第 3 位为 1，则整个输入过程结束。两个状态寄存器的端口地址分别是 0024H 和 0036H，与其相应的数据输入寄存器的端口地址则为 0026H 和 0038H，输入字符分别存入首地址为 BUFF1 和 BUFF2 的存储区中。

```
DATAS SEGMENT
    ;此处输入数据段代码
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    MOV    DI, 0
   MOV SI, 0
BEGIN:
```

```
    IN AL, 24H
    TES AL, 08H             ;查询第一个结束
    JNZ EXIT
    TES AL, 01H             ;查询第一个准备
    JZ BEGIN1
    IN AL, 26H          ;输入数据并存入BUFF1
    MOV BUFF1[DI], AL
    INC DI
BEGIN1:
    IN AL, 36H
    TES AL, 08H             ;查询第二个否结束
    JNZ EXIT
    TES AL, 01H             ;查询第二个准备好
    JZ BEGIN
    IN AL, 38H          ;输入数据并存入BUFF2
    MOV BUFF2[SI], AL
    INC SI
    JMP BEGIN

    ;此处输入代码段代码
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

8-8 给定(SP)=0100H,(SS)=0300H,(FLAGS)=0240H,以下存储单元的内容为(00020)=0040H,(00022)=0100H，在段地址为 0900 及偏移地址为 00A0H 的单元中有一条中断指令 INT 8，试问执行 INT 8 指令后，SP，SS，IP，FLAGS 的内容是什么？栈顶的三个字是什么？

照片



8-11 编写指令序列，使类型 1CH 的中断向量指向中断处理程序 SHOW_CLOCK。

```
DATAS SEGMENT
```

```
    ;此处输入数据段代码
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    MOV    AL, 1CH
    MOV AH, 35H
    INT 21H
    PUSH ES
    PUSH BX
    PUSH DS
    MOV AX, SEG SHOW_CLOCK
    MOV DS, AX
    MOV DX, OFFSET SHOW_CLOCK
    MOV AL, 1CH
    MOV AH, 25H
    INT 21H
    POP DS
    POP BX
    POP ES
    POP DX
    POP DS
    MOV AL, 1CH
    MOV AH, 25H
    INT 21H

    ;此处输入代码段代码
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

第九章 BIOS 和 DOS 中断

9-2 编写一个程序，接受从键盘输入的 10 个十进制数字，输入回车符则停止输入，然后将这些数字加密后(用 XLAT 指令变换)存入内存缓冲区 BUFFER。加密表为:

输入数字: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
密码数字: 7, 5, 9, 1, 3, 6, 8, 0, 2, 4

```
DATAS SEGMENT
    ;此处输入数据段代码
    buffer db 10 dup(?)
    passw  db 7,5,9,1,3,6,8,0,2,4
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
main proc far
    MOV AX,DATAS
    MOV DS,AX
    ;此处输入代码段代码

    mov si,0
    lea bx,passw

loop1:
   mov ah,01h
    int 21h
    cmp al,0dh
    jz exit
    sub al,48
    jb loop1
    cmp al,09h
    ja loop1
    xlat
    mov buffer[si],al
    inc si
    loop loop1

exit:
    MOV AH,4CH
    INT 21H
main endp
CODES ENDS
    END START
```

9-3 对应黑白显示器屏幕上 40 列最下边一个象素的存储单元地址是什么?
对应的存储单元地址是: B000:0F78H

9-5 编写指令把 12 行 0 列到 22 行 79 列的屏幕清除。

```
MOV AL, 0
MOV BH, 07
MOV CH, 12
MOV DH, 22
MOV DL, 79
MOV AH, 6
INT 10H
```

9-6 编写指令使其完成下列要求。

(1) 读当前光标位置

```
MOV AH,3
MOV BH,0
INT 10H
```

(2) 把光标移至屏底一行的开始

```
mov dh,24
mov dl,0
mov bh,0
mov ah,2
int 21h
```

(3) 在屏幕的左上角以正常属性显示一个字母 M

```
mov ah,2
mov dx,0
mov bh,0
int 10h
mov ah,9
mov al,'M'
mov bh,0
mov bl,7
mov cx,1
int 10h
```